1. 为 X 轴上的两个正整数 m 和 n 定义移动操作如下：每次移动操作将 m 向右移动 2 个单位（即 m 的值加 2），将 n 向左移动 3 个单位（即 n 的值减 3）。如果经过若干次移动后他们恰好同时到达某个正整数，返回该正整数。如果无法同时到达某个正整数，返回-1。如果 m 大于等于 n，抛出 string 异常"invalid parameter"。要求编写的函数 func1 的函数原型如下：

```
int func1(int m, int n);
```

测试用例

输入	输出
1, 6	返回 3
10, 9	抛出异常"invalid parameter"
1, 5	返回-1

2. 一个 double 型向量 v 中至少包含一个元素。编写一个函数，找出 v 中与向量所有元素平均值之差最大的元素（可能不止一个）。要求编写的函数 func2 的函数原型如下：

```
vector<double> func2(const vector<double>& v);
```

测试用例

输入	输出
{1.0, 2.0, 3.0}	返回{2.0}
{1.0, 2.0, 4.0, 5.0}	返回{2.0, 4.0}

3. 编写一个函数，删除给定整型向量 v 中的连续重复数，返回结果向量。要求编写的函数 func3 的函数原型如下：

```
vector<int> func3(const vector<int>& v)
```

测试用例

输入	输出
{1, 1, 1}	返回{1}
{1, 1, 2, 2, 1, 1}	返回{1, 2, 1}

4. Circle 类的初始定义如下：

```
class Circle
{
public:
    void setR(double dr) { m_r = dr; }
    double getR() const { return m_r; }
private:
    double m_r;
};
```

请完善 Circle 类的定义，为其提供如下功能：

1) 提供一个带参数的构造函数，用于初始化 Circle 的半径，圆的半径必须大于 0，如果半径小于等于 0，则抛出 string 异常"The radius must be greater than 0!"；

2) 提供一个计算圆面积的成员函数 getArea，计算面积时圆周率π的值取 3.14；

3) 提供一个比较运算符(==)重载函数，比较两个圆的半径是否相等；

4) 将 func4 中注释掉的代码取消注释（但不要增加或删除该函数中的任何代码），并自行确定是否需要增加其它构造函数、成员函数或辅助函数，使得函数 func4 能正确编译并执行。

```
vector<double> func4(double r)
{
    vector<double> info;
    /*
    Circle c1;
    const Circle& rc = Circle(r);
    info.push_back(rc.getArea());
```

```
    Circle c3(r);
    info.push_back(c3.getArea());
    info.push_back( (rc == c3)? 1 : 0 );
    c3.setR(r + 0.1);
    info.push_back( (rc == c3)? 1 : 0 );
    c3.setR(r + 0.000000001);
    info.push_back( (rc == c3)? 1 : 0 );
    */
    return info;
}
```

5. 编写一个函数，输入参数是一个字符串，其中包含若干以空白符（比如空格、制表符、换行符）间隔的整数，按照这些整数的出现次数降序返回它们。如果两个整数的出现次数相等，那么大数在前。要求编写的函数 func5 的函数原型如下：

```
vector<int> func5(const string& msg);
```

你可以使用一个用户自定义类型来记录某个整数及其出现次数，如下：

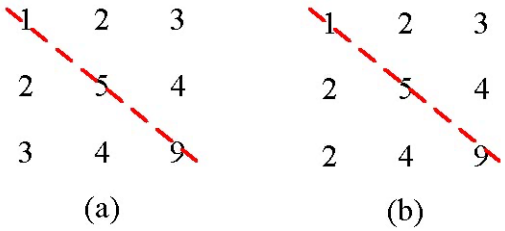
```
struct Record {
    int value; //整数
    int count; //出现次数
};
```

当然，你也可以使用其他方法。

测试用例

输入	输出
"3 3 5 5 5 7 7"	{5, 7, 3}
"3\n 3 5 5 5 7\t 7"	{5, 7, 3}
"1 2 3 4 2 4 4"	{4, 2, 3, 1}

6. 给定一个整型向量，将它的第一个元素到最后一个元素按索引顺序依次排列，如果可以组成一个对称的方阵（即方阵的长与宽相等，以对角线为中心，对应位置的元素相等），则返回 1；如果可以组成一个不对称的方阵，则返回-1；如果不



能组成方阵，则抛出字符串异常"invalid vector"。

比如，给定向量{1, 2, 3, 2, 5, 4, 3, 4, 9},其元素按下标顺序可以组成如图(a)所示的对称方阵，此时应返回 1。又比如，给定向量{1, 2, 3, 2, 5, 4, 2, 4, 9}，其元素按下标顺序可组成如图(b)所示的不对称方阵，此时应返回-1。如果给定向量{1, 2, 3}，显然其元素无法组成长和宽相等的方阵，所以应抛出异常"invalid vector"。

要求编写的函数 func6 的函数原型如下：

```
int func6(const vector<int>& vec);
```